ERDC MSRC PET Technical Report No. 01-22

Improved Parallel Performance for Environmental Quality Models

by

Victor J. Parr
Mary F. Wheeler

14 May 2001

Work funded by the Department of Defense
High Performance Computing Modernization Program
U.S. Army Engineer Research and Development Center
Major Shared Resource Center through

Programming Environment and Training

Supported by Contract Number: DAHC94-96-C0002
Computer Sciences Corporation

Improved Parallel Performance for Environmental Quality Models

Victor J. Parr
and
Mary F. Wheeler
Center for Subsurface Modeling
University of Texas, Austin

1 Introduction

In parallelizing environmental quality modeling (EQM) codes, such as the water quality model CE-QUAL-ICM [1], two of the main bottlenecks are message passing and input/output (I/O). This report describes recent efforts by the University of Texas PET team to improve the parallel performance of CE-QUAL-ICM through the use of more efficient message passing and parallel I/O capabilities.

During a typical water quality simulation, variables such as concentrations, cell volumes, and the time step need to be updated (i.e., passed among processors) within each time iteration. Furthermore, concentrations of dozens of chemical constituents are written to output files at various time steps during a simulation. The updating step was being handled in CE-QUAL-ICM through the use of a "synchronization" point, which involved a collection of MPI [2] (Message Passing Interface) "all-reduce" operations. An analysis of the parallel performance of CE-QUAL-ICM using the VAMPIR tool [3] revealed that this synchronization step was one of the primary bottlenecks in the code. The tremendous volume of output also inhibited parallel scalability.

To remove the synchronization point in the code, persistent, asynchronous sends and receives were investigated for communicating concentrations between MPI processes. A method for localizing the hydrodynamic input files was implemented

in order to eliminate the passing of cell volume information at each time step. Moreover, improved strategies for handling I/O, including MPI parallel I/O tools developed by the University of Tennessee PET team, were investigated.

Below, we describe the implementation of persistent, asynchronous sends and receives, and the use of parallel I/O tools for the improvement of parallel performance of the CE-QUAL-ICM water quality model.

2 Message Passing Code Modifications

CE-QUAL-ICM was parallelized using MPI in previous PET efforts [4, 5]. There are two important aspects of the program which directly affect scalability: the efficiency of the message-passing interface, and the efficiency of the file I/O. VAMPIR was used for profiling CE-QUAL-ICM in a 30-day Chesapeake Bay simulation. This study revealed that a large portion of CPU time was spent communicating the cell concentrations and cell volumes between MPI processes. Significant time was also consumed in MPI all-reduce operations within each time step to determine the next time step. This operation is needed to preserve the numerical stability of the explicit time stepping procedure used in the code.

The communication of the concentrations of active constituents belonging to

ghost cells of neighboring subdomains involves the largest volume of data to be communicated. At every time step, data of size 256,000 bytes is updated per processor. This updating, however, was delayed until the completion of all significant computations within each time step. To reduce this communication overhead, a Fortran90 module was developed to implement asynchronous, persistent message passing in CE-QUAL-ICM. This protocol allows for asynchronous sends and receives of concentrations toward the begining of the time step loop and improves the point-to-point communication performance by overlapping the message passing with computation.

CE-QUAL-ICM takes into account the change of the water levels at each time step. As mentioned above, a VAMPIR profile also indicated that a large portion of CPU time is spent in the communication of the cell volumes at each time step. After a careful analysis of the data dependencies involved in the computation of the cell volumes, two steps were taken for the elimination of the message passing of these volumes. First, the hydrodynamic input file which contains the water level information was localized so that each processor has its own copy at run time. Second, resident and ghost cell volumes were saved into files so that each subdomain contains all the information needed for the computation of cell volumes.

Since CE-QUAL-ICM uses dynamic time stepping, the next computed global

time step needs to be broadcast to every processor. This was accomplished by a collection of MPI all-reduce operations. However, processors tended to get out of phase with each other at the synchronization point while waiting to receive messages. To eliminate this problem, a reduction in the frequency of the MPI all-reduce operations was investigated. The code was modified so that the global time step is updated at every eighth time step intead of at every time step. This change resulted in no discernible differences in the results of the code.

With these code modifications, a speedup factor of 30 was achieved on 32 processors of the Chesapeake Bay EPA simulation. The previous version of CE-QUAL-ICM only gained a speedup factor of 20. These efforts have resulted in researchers completing EQM studies in less than two weeks that previously took a year.

3 Parallel I/O

The original parallel version of CE-QUAL-ICM had an inherent problem in that a large number of files had to be open for storing results. For a typical execution, results were saved into files by appending the solutions of the current time step to the solutions of the previous intermediate time step. For example, the Chesapeake Bay simulation opened 33 files simultaneously per processor. For a 100 MPI pro-

cessor run, the total number of open files would be 3,300. The number of allowable open files is sometimes limited by the operating system. Moreover, a sequential post-processing step was required. The post processor read in two different array mapping files. The first mapping file defined which cells from each array, for each process, are resident cells. The second mapping file indicated where each resident cell, for each process, is assigned to the global array. This process was repeated for each array and for each time step, and was very time consuming for long duration simulations.

A temporary solution to the large number of open files has been implemented by the University of Texas EQM team. This was done by closing all input files after data are read at startup and closing all sequential output files after writing a record, then re-opening the file again using the Fortran APPEND command. With this temporary technique, we reduced the total number of open files from 33 to 13 per processor in the Chesapeake Bay EPA simulation.

For a more efficient solution, the University of Texas EQM team has been working closely with the University of Tennessee PET team in using MPI-I/O for reducing the number of open files in the CE-QUAL-ICM code. As a start, a study of the mapping process has been conducted by the University of Tennessee team. Figure 3 shows an example of the irregular array distribution from this study. Details of the

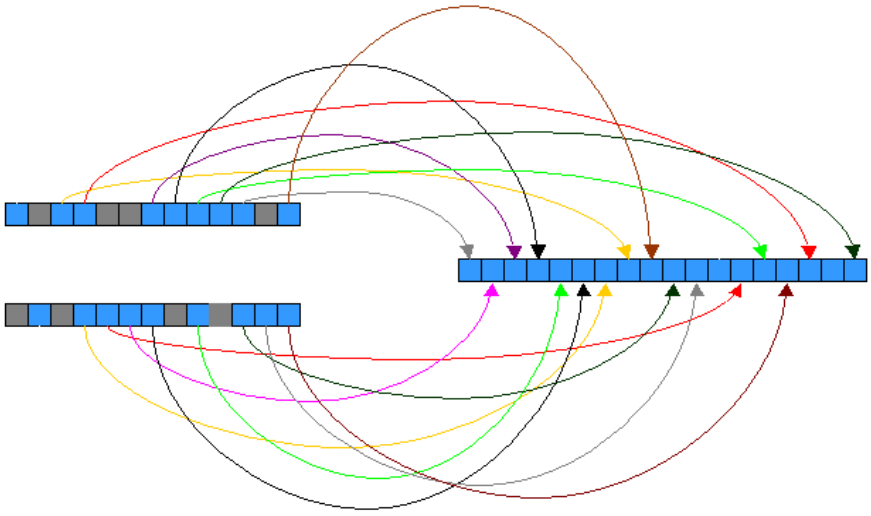

Figure 1: Irregular Data Distribution

implementation of MPI-I/O in the CE-QUAL-ICM code can be found in the PET year 5 technical report [6] from the University of Tennessee PET team.

4 Conclusions and Future Work

We have resolved one of the two main bottlenecks inherent in the CE-QUAL-ICM parallel version. The use of persistent, asynchronous message passing of concentrations improved the scalability of the code. The elimination of the cell volume message passing also makes a significant contribution to this scalability. Current work on implementing MPI-I/O within CE-QUAL-ICM will continue. As a con-

tinuation of this project, the Unversity of Texas EQM team will migrate these techniques into the production CE-QUAL-ICM code.

References

[1] Cerco, C. F. and Cole, T., "User's Guide to The CE-QUAL-ICM Three-Dimensional Eutrophication Model," Technical Report EL-95, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.

[2] Snir, M., Otto, S., Huss-Lederman, S., Walker, D., and Dongarra, J., *MPI—The Complete Reference: Volume 1, the MPI Core*, MIT Press, Cambridge, 1998.

[3] VAMPIR 2.5, "Visualization and Analysis of MPI Programs, Version 2.5," http://www.pallas.de/pages/vampir.htm.

[4] Wheeler, M. F. and Parr, V. " Parallel Sotfware Tools and the Parallel Performance of the CE-QUAL-ICM Water Quality Simulator," Technical Report ERDC MSRC/PET, TR/00-22, March 2000.

[5] Dawson, C.N., Parr, V.J. and Wheeler, M.F., "Issues in parallel computation of flow and transport in surface waters," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, Vol. 1, H.R. Arabnia, ed., Las Vegas, NV, June 26-29, 2000, CSREA Press, pp. 21-27.

[6] Cronk, D., Fagg, G., Moore, S. and Parr, V., "Parallel I/O for EQM Applications," Technical Report ERDC MSRC/PET, TR/01-07, April 2001.